NEURAL NETWORK BASED MODELS FOR FORECASTING

X. DING (1), S. CANU (1,2) and T. DENŒUX (1,2)

(1) Lyonnaise des Eaux / LIAC
Technopolis - Rue du Fonds Pernant
F-60471 Compiègne cedex- France

(2) Université de Technologie de Compiègne – U.R.A. CNRS 817
BP 649 - F-60206 Compiègne cedex - France
e-mail scanu,tdenoeux@hds.univ-compiegne.fr

ABSTRACT

In most industrial systems, forecasts of external demand or predictions of the future system state are necessary to achieve optimal management and control. Forecasting tasks can be formulated as different classes of problems such as function approximation or classification, in which neural network techniques can be applied. In this paper, some examples of forecasting applications using neural networks are presented. Through critical analysis of these applications, our aim is to show the applicability of neural network techniques to forecasting problems, their constraints and limits, and also their advantages and drawbacks as compared to other techniques.

KEYWORDS

Artificial neural networks, Forecasting, Machine learning, Statistical modeling.

1 Introduction

In many industrial systems, it is often necessary to have reliable forecasts of external demand for optimal management and control. Generally, the end-user's needs can be classified in four categories depending on the relationship between the time constant of the underlying system and the forecasting time scale considered. For example, in an economic context, the following types of forecast should be considered:

- Long-term forecasts, corresponding to yearly evolution in economic applications, are needed to design infrastructure and to plan investment. This type of forecast generally has an important economic impact because it provides information for high level decisions.
- Middle-term forecasts are typically issued several months in advance, in order to assist planning of production resources.
- Short-term forecast are often associated with a lead time varying from one day to several hours. Such forecasts are an essential element for optimal control, decision aid in critical situations or detection of abnormal situations.
- Real-time forecasting is concerned with sampling periods not exceeding a few minutes. In this case, the forecasting system performs automatically, and safety becomes a critical issue.

Answers to these specifications can be provided by solving the following forecasting problems:

- Trend detection. The question is to determine whether there exists a trend or not.
- Regularity analysis. In this case, the objective is to model the underlying mechanism producing the time series.

- Irregularity detection. Exceptional events often correspond to crisis situations. The understanding of their underlying causes is often a difficult problem.

If this classification remains valid for a wide range of applications, the time intervals may change depending on the application. For instance, three hours ahead prediction is considered as middle-term forecasting in the radar application presented below. Techniques used to tackle these different forecasting problems are generally based on the following approaches:

- Deterministic modeling, which attempts to describe the physical laws involved by equations. High level prior knowledge and a detailed description of the system structure are needed.

- Conceptual modeling provides a global representation of the system at a macroscopic level. The parameters do not necessarily have a physical interpretation.

- Expert systems are based on expert knowledge about the considered phenomena; this knowledge is often encoded in the form of rules.

- Statistical modeling: in this case, the lack of prior knowledge is compensated by the search for dependencies in empirical data.

Neural network models belong to this last category. They are powerful machine learning techniques which are able to extract the most relevant features from large data sets. This is useful in real-world applications involving complex systems that must be analyzed on the basis of very week prior knowledge. However, an essential requirement for this approach is the availability of a sufficient amount of "good quality" data. It is also important to adopt a well-defined methodology in the development of applications using neural network techniques.

In this paper, we shall present some applications in various domains such as forecasting of water demand, highway traffic, heat power demand, rainfall and urban storm water pollution. These applications have posed several kinds of forecasting problems and have required suitable neural network-based solutions. Through critical analysis of these applications, our aim is to illustrate the general methodology for applying neural networks in forecasting, to highlight their constraints and limits, and to show the advantages and drawbacks of this approach as compared to other approaches.

2 Water demand

Problem Forecasts of daily water consumption are needed to optimize resources for water supply. We make use of real data coming from water distribution networks of different kinds because the causality factors are different for different kinds of water consumption: urban, industrial, or agricultural.

Method of approach The solution provided consists in taking into account the past water consumption, the day of week and meteorological data. The main goal of the application is to avoid large errors, while keeping the mean error within acceptable limits. This application is organized in two distinct phases, the "off line" construction of the neural network, also called the learning phase, and the "on line" use and refinement of the trained network (adaptive phase).

During the learning phase, a feed-forward neural network, also called multi-layered perceptron (M.L.P.) is used to forecast water demand. The synaptic weights are computed so to minimize the mean squared error on the learning set through the back-propagation learning procedure [16]. An important problem in this procedure is to determine when training should be stopped, because an over-trained M.L.P. can have poor generalization performance. This behavior is known as "over learning" [4]. In order to avoid such a drawback, the learning phase is stopped when the generalization error measured on a cross-validation set attains a minimum. After eliminating holiday effects through a linear transformation and normalizing the data, the examples are split into three different sets. The learning set, used to compute the best synaptic weights, the cross-validation set, used to determine when to stop the learning phase and the test set used to evaluate the selected neural network.

Since the water demand time series is not stationary, an adaptive algorithm has to be designed. In order to do so, the effects of the different parameters of the algorithm on the quality and on stability of the solution should be investigated. The following algorithm is proposed:

1. Present a new input vector and make the prediction.

2. Compute the forecasting error.

3. Modify the synaptic weights using backpropagation.

The reliability of the new solution has to be carefully established to avoid trouble due to outliers or other causes. To this end, a supervisor of the whole system is needed. On the one hand, the development of such a supervisor increases the cost of the application, but, on the one other hand, it is necessary to guarantee the system performance.

Results The best results were obtained with four-layer perceptrons. Many four-layer architectures give almost the same results. So, the smallest one is generally used, in order to reduce the learning phase. The architectures used are 20-6-2-1 and 20-6-6-1. A comparison with an ARIMA model showed the superiority of the neural network-based approach [3].

3 Energy demand

Problem The city of Chambery is equipped with a large heat distribution system including power plants and distribution networks. The network of the city of Chambery is 45 km long, with almost 400 heat exchange stations. There are quickly varying weather conditions and all kinds of heating habits. A forecasting system has been designed for optimizing the management of power stations and distribution networks. This system has to manage the available data and to make predictions, taking into account all informative variables such as weather information (sunshine, wind) and sociological information (day of week, holidays).

Method of approach The problem can be approached in two steps: first to determine whether or not a variable has a significant influence on the consumed power, and then to identify and to quantify this effect. Given these specifications, the work is performed the other way around, building first a model including all possible variables and then eliminating the useless ones through sensitivity analysis. The available variables were found to be strongly redundant and a lot of them disappear when using a non-linear model. For economic reasons, the fewer input variables are used, the better the model is. The chosen approach to provide optimal control is to forecast the behavior of the installation. This identification is very difficult to achieve through modeling because of both time delay and complexity reasons. Consequently, a statistical method has to be used. Two different forecasted values are needed by the operator to control the installation: short-term forecasts allow him to control immediately the power station, while middle-term forecasts help him to decide whether or not an additional boiler has to be lighted on. These two different forecasts are performed using different techniques based on neural networks.

Results A back-propagation network has been trained to predict heat demand with a lead time of one to three hours. The network inputs are the heat power consumption during the five past hours, and the typical consumption profile for the next three hours. Data of one heating season (from October 1992 to April 1993) were available for training and testing the system. Performance obtained on the test set, with the neural network (NN) and the "persistence" method which simply consists in taking the last observed value as the prediction, is showed in Table 1.

Methods	1 hour	2 hours	3 hours
Persistence	5.3%	9.2%	12.3%
NN	3.9%	6.4%	8.2%

Table 1: Heat demand forecasting : Mean relative errors on the test set

Methods	from Paris	to Paris
Expert	11.6%	8.7%
NN	8.2%	7.3%

Table 2: Highway traffic forecasting : Mean relative errors on the test set

4 Highway traffic

Problem Middle-term forecasts of daily traffic flow are issued several months in advance, in order to assist the planning of road works and toll-gate management. Short-term forecasts of traffic flow with a lead time of one to three hours are essential in traffic control, the objective of which is to delay (or preferably to avoid) the formation of traffic jams by regulating the access to the network.

Method of approach The proposed approach for middle-term forecasting is to model the relationship between daily traffic flow and calendar and holiday configuration. A multi-layer network was trained using the backpropagation algorithm to perform this task. One major issue of the modeling procedure was the choice and encoding of relevant inputs, which has to be based on prior knowledge about the phenomenon under study. The proposed input coding scheme allows to represent any possible calendar and holiday configurations, which are then mapped onto corresponding traffic flows.

For short-term forecasting, one has the opportunity to make use of traffic flows measured in the past. The central idea of the proposed method consists in looking in the past for traffic situations that are similar to the current one. Different traffic situations were described by a certain amount of prototypes extracted automatically by an unsupervised neural network model (NeoART) based on competitive learning [19]. The method is able to provide not only predictions for the next hours, but also an indication of the associated uncertainty. A very high uncertainty value indicates that the current traffic situation is, to some extent, atypical. This information can help the user to make suitable decisions.

Results Experiments were performed with the traffic data observed at St-Arnoult toll-gate, near Paris [8]. For middle-term forecasting, learning was performed with ten years of data (1982-1991) and the model was tested on the year 1992. Two networks have been trained, one for each direction (from and to Paris). The obtained results have shown that the neural network gives globally better forecasts than those made by an expert (Table 2).

For short-term forecasting, learning was performed using the data of 1990. Predictions with a lead time of one to three hours were made for the year 1991. The obtained prediction results are generally satisfactory, although important errors can still occur in some situations. However, in this case, the uncertainty measure is generally high. Additional data with more examples on the atypical traffic situations will help to reduce this kind of prediction errors.

Discussion Neural networks based models have been proposed to solve two different traffic forecasting problems. In the case of middle-term forecasting, it has been proved that a multi-layer back-propagation network can approximate correctly the complex relationship between daily traffic flow on the one hand, and calendar and school holiday information on the other hand.

In the case of short-term forecasting, the role of the neural network was not to model an input-output relationship, but to create categories among the input data, and to generate prototypes representing each of these categories. The advantage of such an approach is that adaptation to other locations of the highway network is very easy, because the model has only one control parameter (the vigilance parameter). Unlike general-purpose neural networks that perform as black boxes, interpretation or labeling of prototypes is possible. The estimation of uncertainty associated to each prediction is also regarded as an interesting functionality by the end-users.

5 Urban storm water pollution forecasting

Problem During rainfall, the pollution accumulated on impervious areas and in sewer pipes is scoured and carried to waste water treatment plants through sewer systems. In order to limit the damaging effects of this pollution on the environment, treatment plant operators need predictions of the amount of water and pollutants (mainly associated with suspended solids) which are expected to reach the outlet of the drainage catchment. In recent years, significant progress has been made in this area through the collection of experimental data and the development of models for solid transport in sewer systems. However, these models are often impractical to use in real-time because of high computation requirements, need for manual calibration and lack of adaptation. The objective of this study was to develop connectionist models allowing to overcome these limitations.

Method of approach Two non-linear recurrent networks have been developed, for the simulation of rainfall-runoff and solid transport processes [10]. Since only limited amounts of training data are available, prior knowledge has been introduced in these models in the form of specific architectures defining a functional equivalence with a simplified conceptual model [2]. Initial training and on-line adaptation were performed with the back-propagation through time algorithm [15].

Results Experiments were performed on several small-sized urban catchments. Satisfactory results were obtained for both models, despite the scarcity of measurements of rainfall intensity and suspended solid concentration for most events. A simulation example is presented in Figure 1.

Discussion Neural networks are simpler than conceptual models in terms of number of parameters. This simplicity allows fast computation, which facilitates real-time prediction of flow rates and suspended solid concentration. The introduction of prior knowledge in the network architecture design makes training possible with relatively few data. Furthermore, unlike in the case of “black-box modeling” neural networks, most connection weights have obvious physical meaning. The performances of these models are comparable to those of conceptual models, with the advantages of automatic calibration and fast adaptation.

6 Rainfall forecasting using radar images

This work was performed in the context of the Esprit project Neufodi[1]. Two different approaches were investigated, based respectively on rain cell tracking and rain field modeling.

6.1 Rain cell tracking

Problem The existing radar-based rainfall forecasting methods rely on the extrapolation of the estimated advection of rain cells perceived on radar image as reflecting patterns. Advection can be estimated

[1]NEUFODI (NEUral networks for FOrecasting and DIagnosis Applications) , Esprit-II Nr. 5433. Partners: BIKIT, ARIAI, Elorduy Sancho y Cia, LABEIN, Lyonnaise des Eaux; Associated partner : RHEA S.A.

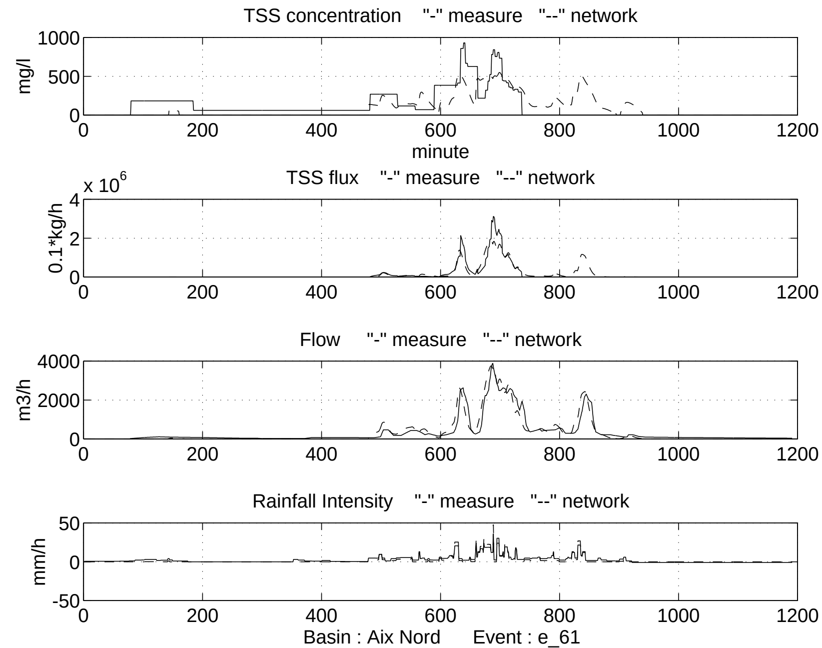

Figure 1: Simulation of a rainfall event with connectionist models.

globally in the whole image [1] or individually for each rain cell [9]. In this last approach, rainfall structures are represented by "echoes" that are extracted and described by geometric characteristics such as size, mass, principle moments of inertia, etc. For estimating the advection of a rain cell, we have to track it in successive images. Many cell matching methods rely on heuristic rules based on cell characteristics and evolution [9]. In our approach, an echo extracted at time t is associated to an echo extracted at time $t + \Delta t$ to form a pair of echos described by a set of feature. The problem can than be formulated as a classification task: each pair has to be assigned to the class of correct matching or to the class of incorrect matching.

Method of approach Neural network based classifiers have been proposed for extracting matching rules from correct and incorrect matching examples [7]. The reference method is the decision tree generation algorithm proposed in [14]. Several approaches were implemented and tested, including (1) using the decision tree to initialize a network; (2) using the decision tree as a feature selector; and (3) training back-propagation networks from scratch, with different architectures.

Results Among the various approaches experimented, the best performance was obtained with a multilayer network presented with a limited number of features selected by the decision tree generation algorithm. The lowest test error rate obtained was 5.9%, which represents a significant improvement over the 8.7% obtained with the decision tree alone. The network constructed independently from the decision tree also showed relatively good classification performance (7.2%).

6.2 Rain field modeling

Problem In the previous approach, rainfall structures were represented by the extracted echo characteristics. This representation depends on the definition of echos, the extraction methods and the characteristics used to describe them. The objective of this alternative approach is to forecast the rain field evolution at different time steps using the raw images at former time steps.

Method of approach The proposed approach is based on the approximation capabilities of artificial neural networks. A new neural network paradigm, called the Competitive Gaussian Potential Function (CGPF) network [6] has been introduced. For each incoming image, the network is trained using the previous weights as an initial state. Learning is achieved through a competitive mechanism, combined with a procedure for adapting the network size to the complexity of the input. Forecast images are obtained by extrapolating the time series of weight values.

Results Experiments performed with radar images corresponding to different types of meteorological situations have shown that the CGPF network performs good approximation of these images. An example of approximation result is presented in Figure 2. For the forecast evaluation, radar images have been divided into 64 regions of 8×8 pixels, each one representing an area of $32 \times 32\text{km}^2$. For each of these areas, predictions of the accumulated rainfall volume for the next 30 and 60 minutes have been performed at several successive time steps. Simulation results have been compared with those obtained with the classical cross-correlation (CC) method [9], showing the superiority of our method. This superiority was found to increase with the forecasting lead time (Table 6.2).

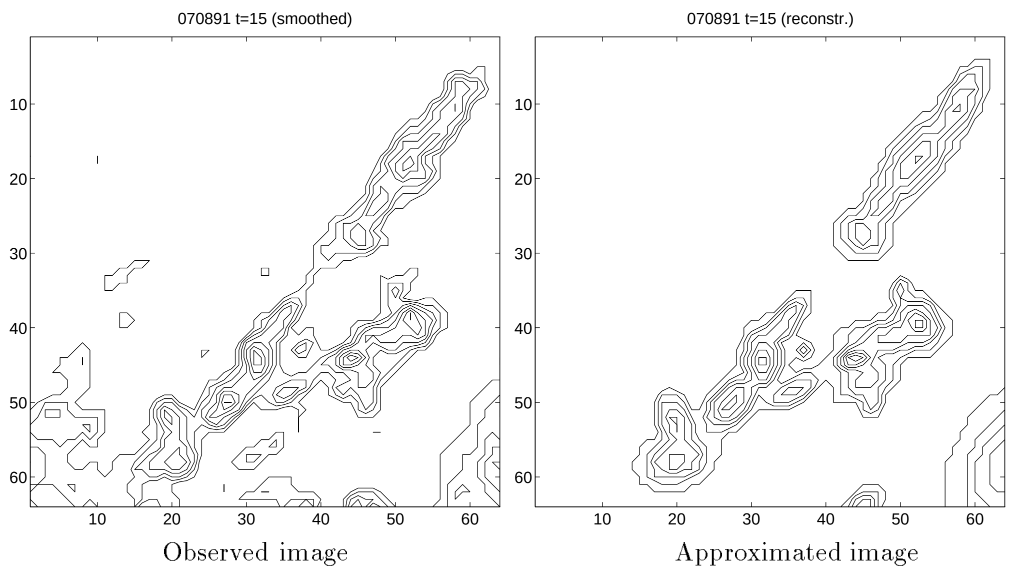

Figure 2: Observed image and approximated image by CGPF Network

7 Towards a methodology for the development of projects using connectionist models

After all the experiments performed, it appears clearly that the solution to a problem and the method to find it depend on the data. However, based on the accumulated experience, the following recommendations may be given as guidelines for future developments. Our approach to Neural Network based applications is based on four phases: development of "reference methods", parameter estimation, performance evaluation, and implementation. Even if the learning phase is often regarded as the most important, the other phases also have some specificity due to the use of neural networks.

	30 mn		60 mn	
	CC	NN	CC	NN
$t_0 = 16$	1.30	0.85	3.88	2.23
$t_0 = 17$	1.11	0.93	3.05	2.19
$t_0 = 18$	1.10	1.00	3.11	2.45
$t_0 = 19$	0.99	0.91	2.94	2.45
$t_0 = 20$	0.92	0.98	3.04	2.58

Table 3: Mean forecasting errors (in mm) — Situation of November 12, 1991

Development of "reference methods" It is always useful to have a basis in terms of results to be improved. We are following in that the recommendations of many authors such as [18] claiming that *"never try a multi-layered model for fitting data until you have first tried a single layer model (linear)"*. Another simple model to deal with is the persistence method which predicts the next values of the time series to be the same as the current one. Note that in some applications (not reported in this paper) the best fitted model was the linear one.

Learning phase When a reference performance criterion is known, the identification of the neural network model can begin. There is not a single algorithm to solve all the problems but the development of the learning phase of the applications presented above has been decomposed in the following subtasks:

1. preprocessing,
2. sample selection,
3. input selection,
4. architecture determination,
5. parameter identification.

In his PhD thesis [13], MacKay writes: *"There are many knobs on the black box of M.L.P. Generally these knobs are set by rules of thumb, trial and error, and the use of reversed test data to assess generalization ability"*. To tackle these different problems, many different techniques may be used mainly depending on the data.

Preprocessing, sample and input selections are not specific issues. But these phases have to be carefully carried out to ensure good forecasting performances. The architecture of the neural network is determined according to two different goals. The objectives of the forecasting algorithm have to be taken into account and a control of the complexity of the solution has to be performed since a "non-parametric" model is proposed. The first constraint determines mainly the number of inputs and outputs, while the second one deals with the number and connectivity of hidden cells. A typical example of architecture determination based on data is the multi-horizon forecast problem. When the objective at time t is not only to forecast some quantity $x(t+1)$ but also $x(t+2), x(t+3), ..., x(t+\tau)$ for some integer $\tau > 1$, three different architectures can be used:

- a single MLP is used to forecast $\hat{x}(t+1)$ and the forecast value is then used as a new input to forecast $\hat{x}(t+2)$, etc.
- a single MLP is used with as many outputs as values to forecast
- τ MLPs are trained, one for each output.

Even if the first approach recommended by [11, 17] gives in most cases the best results, in some other applications the third one has proved more efficient. The determination of the inner architecture of the neural networks has been one of the most fruitful research area of the Neudofi project. As an example of such a constructive method developed within the Neufodi project, the IRO algorithm has been proposed to build a neural network by optimizing the internal representation of the data [12].

Once the architecture has been determined, some tuning of the parameters has to be done. Even if in some techniques such as IRO the two phases are performed simultaneously, some practical tricks are typically useful in parameter estimation. Among them, one can emphasize the following ones:

1. initialization to improve and accelerate this optimization phase; this can be done using decision trees, functional expansion or prototypes [5].
2. back propagation acceleration algorithms
3. pruning methods (model selection) using, e.g., optimal brain damage or regularization approaches such as proposed in [17].
4. selecting the cost function to be minimized,
5. stopping criteria,
6. multiple trials. Choose the best one or use "committee based" decision methods such as the boosting algorithm.

This is only a first guideline of an expected methodology for the use of neural networks but, in order to be able to design it completely, more theoretical work for a deeper understanding of neural networks is needed.

Validation of the approach and performance evaluation Error rates and confidence intervals are needed in most industrial applications. In order to estimate these quantities, the validation methodology is quite similar to the "classical" one used in other statistical projects. The main difference in the validation phase is due to the nature of the example-based programming methods. The nature of the prior hypothesis made when using neural networks provides a local method within the domain of training data. It is the reason why the determination of a validity domain is so important in neural network applications. The consequence of the existence of a validity domain on software development is the necessity to develop a preprocessing module whose task is to determine whether a new input belongs or not to the validity domain. Another reason for using a supervisor is the plasticity-stability dilemma: since the parameters are tuned in an adaptive way, this adaptation schema has to be monitored.

8 Conclusion

Neural network models allow to build efficient forecasting applications. But the development method has to rely on sound statistical principles. A simple learning phase can take quite a long time, and the determination of a relevant set of parameters may require many trials. Therefore, it is not sure that a neural network application can be developed in shorter time than a usual method, as it is currently claimed.

We pointed out that an independent cross-validation set had to be used to have a good stopping criterion. About the adaptive phase, we notice that there may exist an optimal adaptive step, or at least an "optimal step size region". The consequence of such a result is that it is no longer interesting to test all the algorithms but the relevant question becomes: when to use such algorithm ? In other words, the important point is to determine on what kind of problem such or such algorithm is expected to yield good results. It is the reason why it is important to be able to classify the different kinds of problems according to some criterion to be determined.

Furthermore, the examples presented in this paper have demonstrated that there is no universal neural network paradigm suitable for all kinds of forecasting problems. For each problem, a detailed analysis of domain data and the acquisition of prior knowledge are necessary to find a suitable connectionist model. Although multi-layer back-propagation networks are the most commonly used, training from scratch with all possible inputs and all available raw data has often proved ineffective. According to our experience, the introduction of prior knowledge in input selection, input encoding or architecture determination is often very useful, especially when the available domain data is limited. Another important point is that, in practice, users often need some explanation or indication of the uncertainty associated to a prediction for making their decisions. From this point of view, general-purpose black-box neural network models are not as convincing as networks having interpretation possibility.

References

[1] G. L. Austin and A. Bellon, *The use of digital weather radar records for short-term precipitation forecasting*, Quarterly Journal of the Royal Meteorological Society, 100 (1974), pp. 658–664.

[2] J.-L. Bertrand-Krajewski, *A model for solid production and transport for small urban catchments: Preliminary results*, Water Science and Technology, 25 (1992), pp. 29–35.

[3] S. Canu, R. Sobral, and R. Lengellé, *Formal neural network as an adaptative model for water demand*, in Proceedings INNC'90, Kluwer Academic Publishers, 1990, pp. I–131–136.

[4] Y. Chauvin, *Dynamic behavior of constrained back-propagation networks*, in Advances in Neural Information Processing Systems 2, D. Touretzky, ed., Morgan Kaufmann, San Mateo, CA, 1990, pp. 643–649.

[5] T. Denœux and R. Lengellé, *Initializing back-propagation networks with prototypes*, Neural Networks, 6 (1993), pp. 351–363.

[6] T. Denœux and P. Rizand, *Analysis of radar images for rainfall forecasting using neural networks*, Neural Computing and Applications, 3 (1995).

[7] X. Ding, T. Denœux, and F. Helloco, *Tracking rain cells in radar images using multilayer neural networks*, in Proceedings of ICANN'93, S. Gielen and B. Kappen, eds., Springer-Verlag, London, 1993, pp. 962–967.

[8] X. Ding, R. Lengellé, T. Denœux, and C. Ulbricht, *Traffic forecasting application*, Tech. Rep. D/501.3/LY01/01, Neufodi, Esprit Project, 1993.

[9] T. Einfalt, T. Denœux, and G. Jacquet, *A radar rainfall forecasting method designed for hydrological purposes*, Journal of Hydrology, 114 (1990), pp. 229–244.

[10] N. Gong, X. Ding, and T. Denœux, *Urban stormwater pollution forecasting using recurrent neural networks*, in International Conference on Engineering Applications of Neural networks (to appear), Helsinski, Finland, August 1995.

[11] A. Lapedes and R. Farber, *Nonlinear signal processing using neural networks: Prediction and system modelling*, Tech. Rep. LA–UR–87–2662, Los Alamos National Laboratory, Los Alamos, NM, 1987.

[12] R. Lengellé and T. Denœux, *Optimizing multilayer networks layer per layer without back-propagation*, in Artificial Neural Networks II, I. Aleksander and J. Taylor, eds., North-Holland, Amsterdam, 1992, pp. 995–998.

[13] D. J. C. MacKay, *Bayesian Methods for Adaptive models*, PhD thesis, California Institute of Technology, La Jolla, CA., 1992.

[14] A. Neumann, *Introduction d'outils de l'Intelligence Artificielle dans la prévision de pluie par radar (In French)*, PhD thesis, Ecole Nationale des Ponts et Chaussées, Paris, 1991.

[15] S. W. Piché, *Steepest descent algorithms for neural network controllers and filters*, IEEE Transactions on Neural Networks, 2 (1994), pp. 198–211.

[16] D. Rumelhart, G. Hinton, and R. Williams, *Learning internal representations by error propagation*, in Parallel Distributed Processing, D. Rumelhart and J. McClelland, eds., MIT Press, Cambridge, MA, 1986.

[17] A. Weigend, D. Rumelhart, and B. Huberman, *Generalization by weight-elimination with application to forecasting*, in Neural Information Processing 3, R. Lippman, J. Moody, and D. Touretzky, eds., Morgan Kaufmann, San Mateo, CA, 1991, pp. 875–882.

[18] M. S. Y. Hagahsi and S. I. Gallant, *Multi layer vs single layered neural networks and an application to reading hand stamped character*, Neural Networks, 2 (1990), pp. 916–930.

[19] H. Yin, R. Lengellé, and P. Gaillard, *NeoART : une variante du réseau ART2 pour la classification*, Proceedings of Neuro-Nîmes'90, (1990), pp. 167–169.